# LArSoft/LArLite Interoperability Project Plan


# Chris Jones, Marc Paterno



Version 0.1 2015/10/22


## Use Cases

### Case 1

Developer using LArLite wants to use LArSoft data products. NOTE: this explicitly excludes accessing LArSoft algorithms from LArLite.

Solution: The LArLite build system will use defined environment variables to get to libs and headers from all necessary dependencies. The cross use of data products requires that both LArLite and LArSoft be using ROOT 6. Not all LArSoft data products will be appropriate for use in LArLite. The LArLite and LArSoft communities will need to decide on the common subset of data products to be used. The common subset of data products will have to be in a LArSoft repository (or repositories) which is independent from the remaining data products.

### Case 2

Developer using LArLite wants to allow others to use this code in LArSoft. He wants to keep developing using LArLite, not LArSoft.

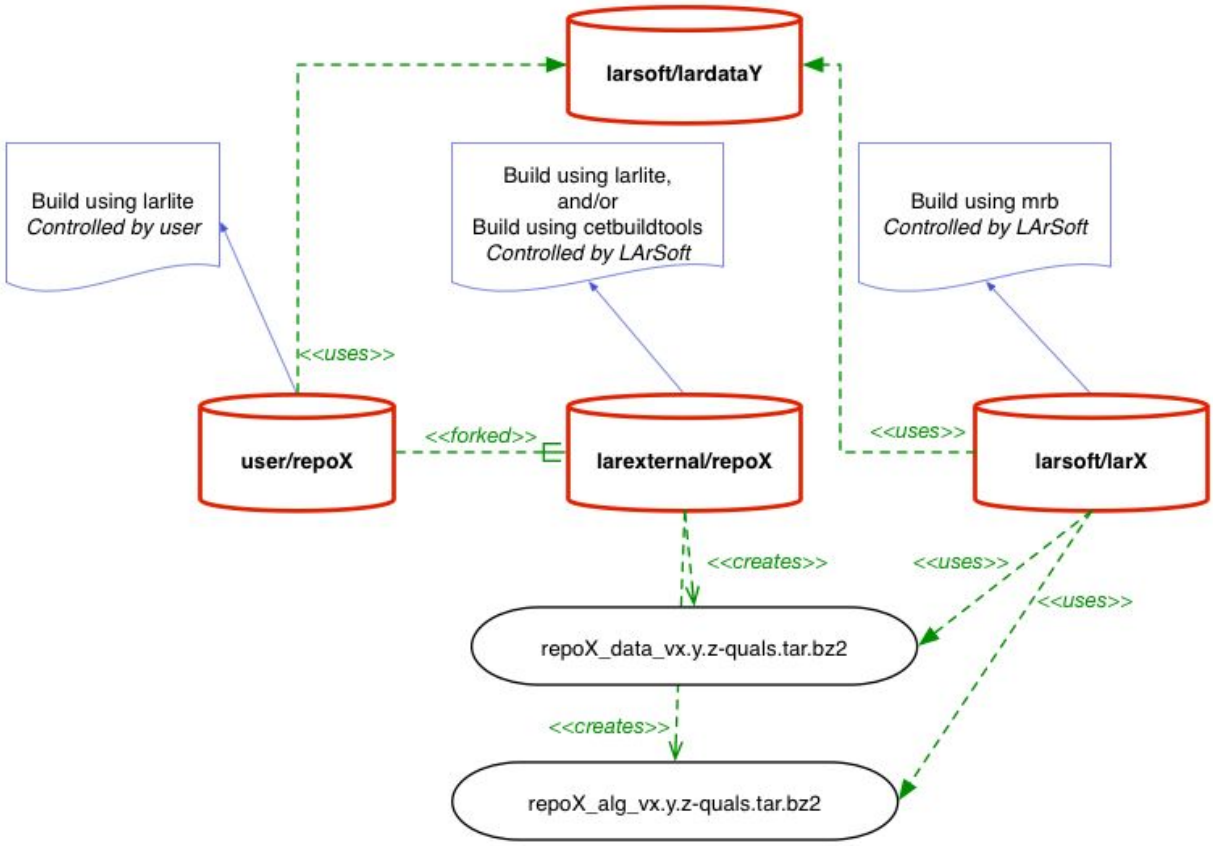
Solution: LArSoft will create and maintain a forked version of the developers git repository. The forked git repository will be used to create UPS products which can then be used by LArSoft code. See *Repository Relationship* section for further details. This mimics how LArSoft presently interacts with externals such as ROOT, GEANT4 and Pandora.

### Case 3

uBoone scientist wants to use python or ROOT to produce plots from a LArSoft production *art*/ROOT data file. There should be no need to create a different file format to support this.

Solution: A artlite::Event class which can read an art/ROOT data file and give access to the data for an event in a manner similar to how data is obtained from within an *art* module.

# Repository Relationship




user/repoX is the developer's personal git repository which holds the code she has developed and now wants to share with the LArSoft community.

When the LArSoft community agrees to include the code the LArSoft group then forks the user/repoX to create larexternal/repoX. Within the new larexternal/repoX repository the LArSoft group creates a new branch containing the version of the code LArSoft wishes to use. The branch also contains newly added cetbuildtools files to allow the production of the UPS build products (repoX_data_vx.y.z-quals.tar.bz2 and repoX_alg_vx.y.z-quals.tar.bz2) and to allow MrB to properly interact with the repository. The LArSoft created branch can also exclude any files from the original user/repoX repository which are not required for use in LArSoft. The LArSoft created branch can also contain modifications to the code which are not in the user/repoX in order to handle needed bug fixes or feature requests. Changes to either the original user/repoX or the larexternal/repoX can be transferred to the other repository using the standard git tools. The LArSoft build system will use the cetbuildtools management files in larexternal/repoX to create UPS build products which can then be used by code within the standard LArSoft code repositories.

# Assumptions

Trying to reduce the dependencies of the ROOT UPS product is not part of this project.
We require a LArLite user to contribute to testing of this system.

# Deliverables

- Demonstrator of software solving each use case. This may include changes in LArLite and/or LArSoft which are contained in a project forked git repository.
- Documentation for how to do the tasks, rules to follow, etc for using the demonstrator and to show how to extend these techniques for other code.
- Changes to *art* necessary to support use cases.
- Changes to nutools necessary to support the cases.
- Policies for maintenance of the repositories.